NordDesign 2014
August 27 – 29, 2014
Espoo, Finland / Melbourne, Australia

Software support for the consistent transition from requirements to functional modeling to system simulation

Fabio Dohr[1], Boris Eisenbart[2], Christian Huwig[1], Lucienne Blessing[2], Michael Vielhaber[1]

[1]Institute of Engineering Design, Saarland University, Germany
dohr@lkt.uni-saarland.de
vielhaber@lkt.uni-saarland.de
[2]Engineering Design and Methodology Group, University of Luxembourg, Luxembourg
boris.eisenbart@uni.lu
lucienne.blessing@uni.lu

Abstract
This paper addresses the realization of a consistent modeling chain from requirements modeling to function modeling and eventually system modeling intended to support early system simulation. Therefor, a prototypical software support has been developed, which links existing software tools for requirements management, function modeling and system simulation and enables a consistent transition of information between them. The paper illustrates the development and initial evaluation of the prototypical software support and discusses potentials for further development and improvement.

Keywords: Requirements, function modeling, system simulation, model consistency

1 Introduction

Conceptual design of technical products encompasses the central transition from a problem to (alternative) solution concepts, which are intended to fulfil the requirements and functionality expected from the product. Conceptual design is considered to be the most influential stage for the design of a product and, therefore, to be requiring a joint effort of the involved designers [1]. Particularly in interdisciplinary design, the success of such collaborative work depends on the ability of designers to share their concepts and ideas [2], as well as to establish a shared understanding of the system under development – including its requirements, expected functionality, and solution elements to be implemented – across all involved disciplines [3]. Technical products can be composed of mechanical, electrical or software systems or can be a combination of these, i.e. a mechatronic system. However, the introduction of new types of systems in industry, such as “Product-Service Systems” (hereafter PSS) that integrate services with technical products, extends interdisciplinary development by including further disciplines. The solution space is hence particularly large and different combinations of diverse solution elements – which may be employed separately, in combination or in exchange for one another – may fulfil the requirements and expected functionality (see e.g. [4]).

From a modeling point of view, the conceptual design stage mainly encompasses requirements specification, function modeling, and models representing the potential solution

concept [5]. Consistent transition of information about the requirements, the expected functions, and solution elements into early system simulation may provide suitable means for supporting the comparison and selection of viable solution concepts [6]. Currently available approaches for modeling the different information are mainly discipline-specific and to a large extent disconnected from one another (see [7]). Only few approaches (see e.g. [8, 9]) strive to support integration of different disciplines and a consistent transition of modelled information during conceptual design. However, the support offered by these approaches is limited thus far, as they provide insufficient support for modeling information relevant to designers from all disciplines typically involved in the development of mechatronic systems or PSS [10].
The research presented in this paper focuses on developing a prototypical software-based modeling support, which can realize the required consistent transition of information from the requirements to early system simulation bidirectionally. The developed modeling support uses the recently proposed Integrated Function Modelling (IFM) framework [11, 12] as basis. The IFM framework was selected as it provides a modular, adaptable function modeling approach that relates between contents prominently addressed in function models from different disciplines, and further provides links to system structural modeling and established tools for system simulation (as shown in [7]). The IFM framework expands the scope of function modeling to address users and peripheral technical artefacts interacting with the system under consideration in different use cases, as well as related entities, such as system states, operand states, and the bidirectional impacts between system elements fulfilling the required functionality.

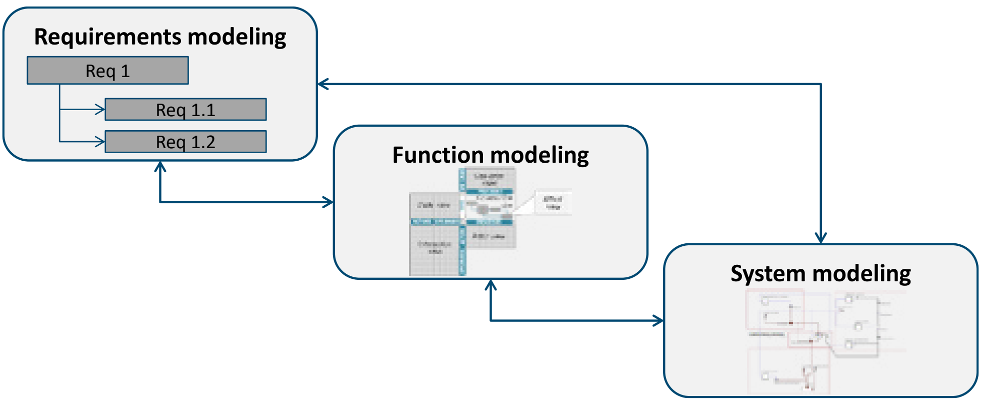

Figure 1: Chain from requirements modeling to function modeling to system modeling

The paper illustrates the development of the targeted prototypical modeling support. The developed support combines different existing software tools and realizes the bidirectional exchange of information between them. The development was part of a 10-months research project collaboratively performed at the authors' research institutions. The following chapter presents the requirements for the developed software support and the selection of the tools that will be used for modeling the requirements, function modeling with the IFM framework, and selection of the an interdisciplinary simulation tools. Chapter 3 then proceeds to illustrate how these different tools are combined in the developed software prototype, before moving to the initial evaluation in Chapter 4. Finally, the results and their implications are discussed.

2 Analysis and choice of software tools

In this section, suitable tools for each of the three steps depicted in Figure 1 are analyzed. Regarding each of the three steps depicted in Figure 1, a vast amount of software tools is available. In order to select one tool for the generation of a consistent tool chain, requirements are defined. At first, this is done for the overall tool chain and based on that for each of the three steps.

2.1 Requirements on an overall tool chain

For the overall tool chain the following requirements are essential:

- Global consistency of the steps depicted in Figure 1
- Bidirectional link of the individual tools
- Automated exchange of information and data through the interfaces
- Tool-spanning traceability of requirements and changes
- Tool-spanning integration of information from function modeling

As indicated before, there are a few tools already available which try to implement the desired tool-chain and thus the consistent transition of information addressed in this paper, e.g. IBM Rational Harmony, CATIA Systems or SysML-based tools. Nevertheless, none of them fulfills all the requirements in order to provide a holistic tool chain for early conceptual phases from requirements to function modeling to system simulation with bidirectional linking.

2.2 Requirements modeling tools

Based on the analysis of several tools and the overall tool chain requirements, the following requirements on requirements modeling tools are used to select suitable software tools:

- Possibility of use-case specific classification of requirements
- Traceability of requirements
- Support of change management
- Ease of use
- Possibility of adaptions of the software
- Providing interfaces and compatibility to other software tools

In a comprehensive study, numerous tools for requirements modeling were considered and assessed regarding their suitability. An overview is provided in [13]. Since it would exceed the scope of this paper, there are only a few of these tools presented in following: one tool which is complimentary, one high-end tool, one niche product with specialized functionalities and one tool with a large user group. The results of this assessment are depicted in Table 1.

Table 1: Assessment of different tools for requirements modeling
(X: fulfilled, (X): partly fulfilled, no entry: not fulfilled)

Tool	Possibility of use-case specific classification of requirements	Traceability of requirements	Support of change management	Ease of use	Possibility of adaptions of the software	Providing interfaces and compatibility to other software tools
Microsoft Excel/Word	X		(X)	X		(X)
ProR	X	X		X	X	
CaliberRM	X	X	X			(X)
Polarion Requirements		X	X			(X)
IBM Rational DOORs	X	X	X		(X)	X

The assessment shows that IBM Rational DOORs fulfills most of the requirements. The only limitation concerns its handling (ease of use), which results from its exceptionally high functionality. Because of this high functionality and particularly because of the numerous interfaces provided to other software tools, IBM DOORs has been selected for the prototypical implementation of the software tool chain.

2.3 Function modeling tools

In the following, the requirements on a software tool for the implementation of the IFM framework are listed, which have particularly been adapted from Gausemeier et al. [8]:

- Holistic description of the system with the IFM framework
- Model consistency
- Intuitive (graphical) modeling
- Interfaces to requirements modeling
- Interfaces to system simulation
- Possibility of adaptions of the software
- Ease of use

In contrast to requirements modeling tools, the number of available tools is rather small. Many of the tools are not commercial but originate from research projects.

As depicted in Table 2, IBM Rational Rhapsody is the only tool which can model the entire system using the IFM framework, which sets it apart from the other tools considered. Moreover it fulfills the remaining requirements (despite slight limitations regarding adaptability) and offers sophisticated interfaces to IBM DOORs, due to their common origin in the IBM Rational family. Therefore IBM Rhapsody is used for function modeling.

Table 2: Assessment of different tools for function modeling
(X: fulfilled, (X): partly fulfilled, no entry: not fulfilled)

Tool	Holistic description of the system with the IFM framework	Model consistency	Intuitive modeling	Interfaces to requirements modeling	Interfaces to system simulation	Possibility of adaptions of the software	Ease of use
Cambridge Advanced Modeller			X			X	X
FunctionCAD			X	(X)		X	X
Microsoft Visio			X	(X)			X
Matlab/Simulink		X	X	X	X		
IBM Rational Rhapsody	X	X	X	X	X	(X)	X

2.4 System modeling tools

In the context of this work system modeling is intended to enable simulation which is done on system level covering aspects of several engineering domains. Furthermore, simulation always requires a computable model [14]. Based on these considerations, the following requirements can be derived:

- Quick and easy workflow
- Quick transfer of (mental) models into computable code
- Comprehensive description of multi-domain system
- Bidirectional interfaces to the two preceding steps
- Integration of information from function models

For system modeling mainly three modeling languages have become established: Modelica, VHDL-AMS and SysML. Therefore, tools using these modeling languages are considered in this research. In addition, Matlab/Simulink as a separate tool has become established for such purposes as well and is therefore also considered, as depicted in Table 3.

Table 3: Assessment of different tools for system modeling and simulation
(X: fulfilled, (X): partly fulfilled, no entry: not fulfilled)

Tool	Quick and easy workflow	Quick transfer of models into computable code	Comprehensive description of multi-domain system	Bidirectional interfaces to the two preceding steps	Integration of information from function models
MagicDraw (SysML)	X	X	X	(X)	(X)
Matlab/Simulink		X	X	X	(X)
ANSYS Simplorer (VHDL)		X	(X)		(X)
Dymola (Modelica)	X	X	X	X	X
ITI SimulationX (Modelica)	X	X	X	X	X

Compared to tools for requirements modeling and function modeling, there are fewer differences between the capabilities of the simulation tools. Nevertheless, Modelica-based tools seem to have additional benefits, particularly regarding the integration of the information provided by the IFM framework. Since the authors have experience with SimulationX, it is chosen for simulation purposes.
Based on the described assessment, for each of the steps depicted in Figure 1, specific software tools have been selected: IBM Rational DOORs for requirements modeling, IBM Rational Rhapsody for function modeling and ITI SimulationX for system modeling and simulation. In the following chapter, the concept of the developed software prototype is described, which is intended to realize a consistent transition of information between the individual tools.

3 Concept of a software prototype

The basis for the software prototype is built by the choice of the three software tools which are to be linked. The interfaces should provide bidirectional linking of the individual artefacts. For this it is important to have a software interface which is used to exchange information but it is also essential to define the methodological interface – which might be the larger and more difficult part. In Figure 2 the overall tool chain including interfaces is depicted. In the following those interfaces between the artifacts are discussed.

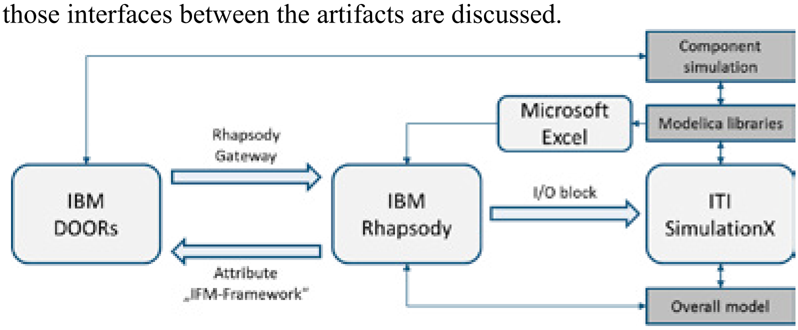

Figure 2: Overall tool chain including interfaces

3.1 Interface: requirements modeling – function modeling

The IT part of the interface between requirements and function modeling is relatively easy to generate: both selected software tools already have a sophisticated, bidirectional interface implemented, called “Rhapsody gateway”. This means that requirements can be assigned to entities of the IFM framework in Rhapsody – and are also updated – from both sides. From a methodological point of view, it is essential to know which information from the requirements are relevant and necessary for function modeling and to which of the entities addressed in the IFM framework they refer to. In turn, this also means that information which has been gained during function modeling will have impact on requirements – whether they are updated or new requirements are derived. The bidirectional link is created by an attribute “IFM framework” which assigns the relevant requirements to entities of the IFM framework. This can also be seen in Figure 3 in the following chapter. In this way the relevant requirements can also be visualized in the appropriate view of the IFM framework.

3.2 Interface: function modeling – system simulation

This interface is more complex from both the IT view and the methodological view. Following function modeling, further steps for concretization are necessary in order to enable system simulation. The generation of a system model requires a certain degree of detail, e.g. a basic geometry and the definition of material. Basically this can be done in two ways: either by concretizing the system concept within the IFM framework, which requires a certain degree of knowledge about the system. Otherwise, a separate step is required which is used to find solution principles to the individual entities of the IFM, e.g. a morphological matrix as depicted in Figure 2. The IT link is done via a hyperlink in both cases. For each entity of the IFM framework which is relevant for system simulation, an empty I/O block is generated via a hyperlink in SimulationX. Those blocks are filled with models representing the solution elements selected in the IFM framework. One feature of Modelica-based tools are model libraries. In this case, those libraries can be used as design catalogues in order to fill the empty I/O blocks. In parallel to that, a morphological matrix can be used to assist the generation of overall system concepts [15]. This matrix is also used as a link back to Rhapsody: the solution elements are exported from the matrix and assigned to the individual entities in Rhapsody. Once the simulation shows that a concept does not fulfill the requirements, the matrix and at the same time the function model in Rhapsody are updated.

3.3 Interface: requirements modeling - system simulation

The third link is essential for validation purposes. In the requirements model the *required* properties[1] of the system to be developed are defined, while the simulation is used to determine the *actual* properties of the system. By a comparison of both tools, conclusions can be drawn regarding the suitability of the developed concept for the specific development task. Therefore, the requirements have to be linked to system simulation. But also parameterization of the system model benefits from this link: through the assignment of requirements, parameters can be directly derived.

Since the integration of DOORs and SimulationX on software level is limited thus far, the link between those two tools is rather difficult to establish. Currently this is done through attributes in DOORs, which define which requirements have to be simulated and which requirements are used to parameterize the model. The link back from SimulationX to DOORs has not been implemented yet.

---

[1] In this context the definition of properties according to Weber et al. [16] is used.

4 Evaluation

In this section the basic approach is validated on the example of a coffee vending machine. This example has already been used by Eisenbart et al. for validation purposes in relation to the IFM framework. A detailed description can be found in [10, 12].

Starting according to Figure 1 with the definition of requirements, those are managed with DOORs as exemplified in Figure 3.

ID		IFM-Framework	SimulationX	Simulation	Evaluation
HL_Req_1	**1 Technical requirements**				
HL_Req_12	**1.1 Environment**	Use-Case			
HL_Req_13	Maximum noise generation during operation must be below 65 dB	Actor Process			
HL_Req_14	Mode for reduced energy consumption must be implemented	Actor Process			
HL_Req_17	Energy consumption Operation: < 35Wh Standby: <1W	Actor Operand			
HL_Req_15	**1.2 Safety**	Use-Case			
HL_Req_27	**2 Main functionalities**	Use-Case			
HL_Req_29	Fully automatic preparation of coffee				
HL_Req_30	Time for coffee prepartion <10 min	Actor Process	X		fulfilled
HL_Req_33	Sales price <300 Euro	Actor			
HL_Req_34	Autonomous maintenance	Actor Process			
HL_Req_6	**3 Comfort**	Use-Case			
HL_Req_7	Autonomous descaling	Actor Process			
HL_Req_8	Choice of quantity: 1-10 cups (100-1000 ml)	Actor Process Operand	X	heat flow = 1.5° C/min	fulfilled

Figure 3: Excerpt of the requirements model within DOORs

The different attributes which are depicted on the right side of Figure 3 are used as bidirectional links to the following stages. In this specific case, the individual requirements are mapped to the different views of the IFM-framework through the corresponding attribute. Hence, using the Rhapsody gateway, requirements can be directly shown in the IFM views. This is exemplified on the right side of Figure 4 for an excerpt of the process flow view.

Regarding simulation, e.g. the time for the preparation of a coffee has to be verified. In this case, the attribute "SimulationX" has to be set. The attribute "Evaluation" serves a link back from SimulationX to DOORs. In the example of Figure 3, the simulation shows that the requirement is fulfilled.

The implementation of the IFM framework within Rhapsody can be easily done with some minor adjustments of the structure of the IFM framework. The model tree as an overview of all IFM views is depicted on the left side of Figure 4.

Based on the entities defined within the IFM framework, a hyperlink is set from Rhapsody to SimulationX which defines an empty I/O block in the simulation model, corresponding to the IFM entity. Those are represented in Figure 5 by the boxes. Using model libraries those empty boxes can be filled with solution principles. At this point the morphological matrix from Figure 2 can be used. A benefit of the use of model libraries is that the simulation model of the entire system is more or less automatically generated.

At this point the model has to be parameterized which is supported by the attribute "Simulation" in DOORs where information which is relevant for simulation and modeling is marked. Based on the simulation results the link back to the requirements is done through a comparison of requirements and simulation results, which is currently done mainly manually.

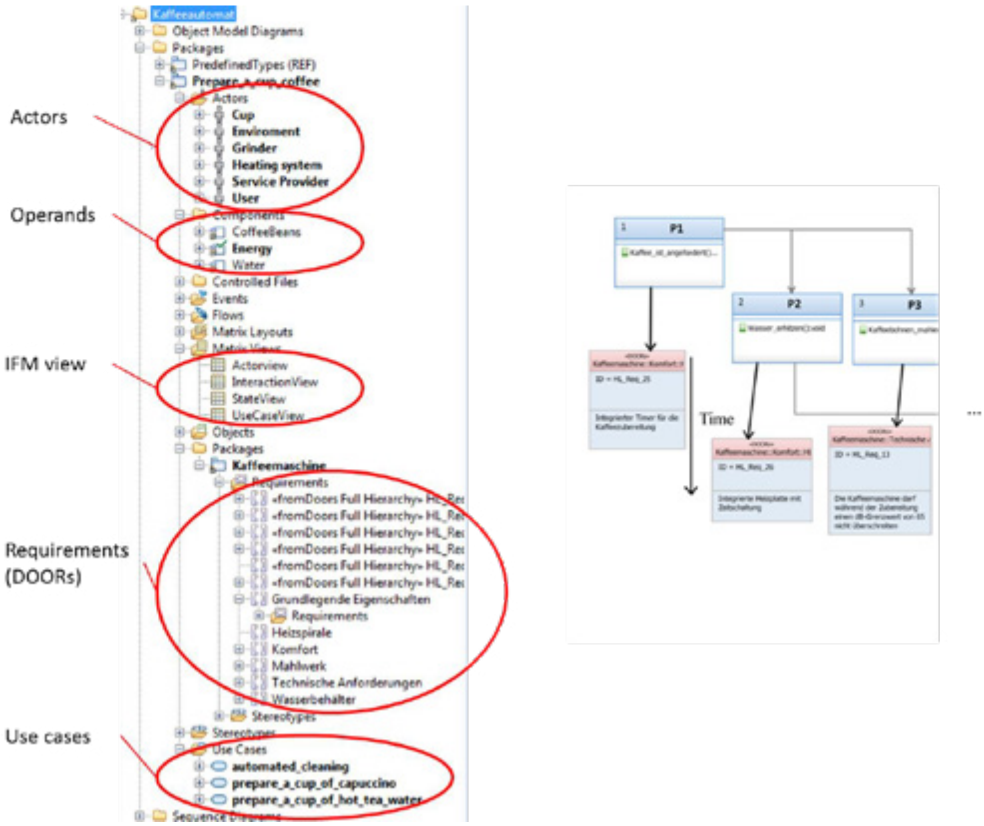

Figure 4: Left: Model tree for the IFM model in Rhapsody
Right: DOORs – Rhapsody link on the example of the process flow view

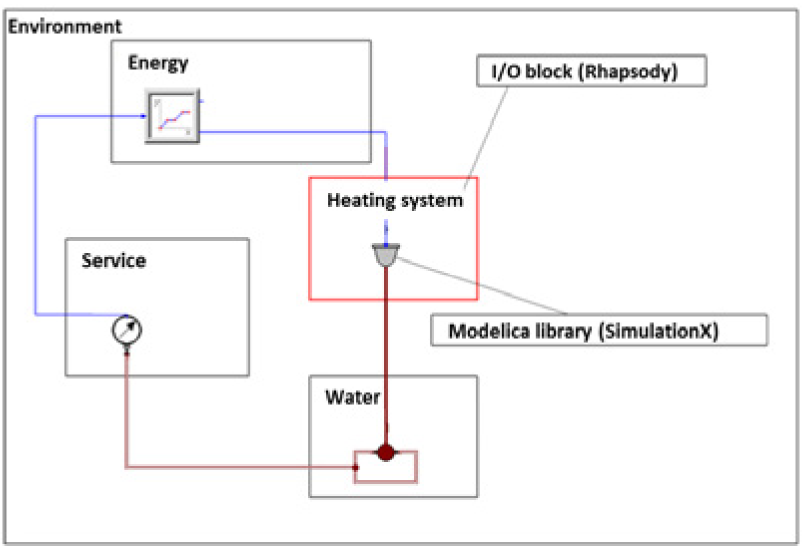

Figure 5: Example of a simulation model generated in SimulationX from IBM Rhapsody

5 Discussion

In this paper the development of a software prototype is described, which is intended to enable a consistent and bidirectional link between requirements modeling, function modeling and system modeling.

The bidirectional link between requirements and function modeling has been successfully implemented by using the IBM Rhapsody gateway. In this way it is possible to link requirements from IBM DOORs and the individual entities of the IFM framework, which is modeled using IBM Rhapsody. By combining both the IBM tools, the link between requirements modeling and function modeling is even automated to a large degree.

The link between IBM DOORs and ITI SimulationX, which is used for system modeling and simulation, as well as between IBM Rhapsody and ITI SimulationX has also been implemented. However, these particular links and the bidirectional transformation of information have to be performed mainly manually by the designers thus far. Especially the link back from simulation to the preceding steps is not yet fully automated. Nevertheless, the basic links have been successfully implemented and future endeavors will focus on further improvement of the automated transition of information.

Although the software prototype requires further improvements, this work shows that from a software point of view, linking of the individual modeling steps can be realized by adequately linking currently available software tools in the developed prototype. While linking the different modelling steps has been realized what is missing is the methodological support or guidance for the designers regarding the transition of information between those individual steps.

For linking of the modeling steps, and the hence linking the inherent information, it seems far more important to consider methodological and process aspects. The validation example has shown that the decision on which information has to be linked and in which particular way will require considerably more attention in the future than connecting the software tools as such. At that point, there is a lack of guidance and several questions arise, which need to be addressed:

- Which requirements influence which subsequent steps and in which way?
- Which information from function modeling and system simulation affect requirements?
- Which information and requirements have to be considered in the IFM model? In which entity of the IFM does this have to be modeled?
- Which information is needed for simulation and where does this come from?
- What happens if there are significant changes in one or more of the models or the system as such?

6 Conclusion

In this paper the consistent chain of information from requirements modeling to function modeling to system modeling and simulation is addressed by the development of a software prototype bidirectionally linking existing software tools. The application of the prototype has been demonstrated using a validation example.

The main finding of this work, supported by the application of the prototype on the validation example, is that the consistent and bidirectional linking can be implemented by linking existing software tools adequately. The main problems have occurred during deciding which information has to be transferred between the different modeling steps and how. Therefore, processes and methods, which are required to support designers in transferring information between the individual modeling steps, may yield a large benefit for improvement of conceptual design. These aspects will have to be comprehensively addressed in future research. Furthermore, based on the insights gained from this research the software prototype will also be refined and improved.

References

[1] Chakravarthy, B.K., Albers, A. and Schweinberger, D. (2001) 'Collaborative Environment for Concept Generation in New Products', Proceedings of International Council of Societies of Industrial Design (ICSID 2001).

[2] Buur, J. and Andreasen, M.M. (1989) 'Design Models in Mechatronic Product Development', Design Studies, Vol. 10 (3), pp. 155-162.

[3] Kleinsmann, M. (2008) 'Barriers and Enablers for Creating Shared Understanding in Co-design Projects', Design Studies, Vol. 29 (4), pp. 369–386.

[4] Müller, P., Schmidt-Kretschmer, M. and Blessing, L.T.M. (2007), 'Function Allocation in Product-Service-Systems – Are there Analogies Between PSS and Mechatronics?', Proceedings of the AEDS Workshop.

[5] Eisenbart, B., Gericke, K. and Blessing, L., (2011) 'A Framework for Comparing Design Modelling Approaches Across Disciplines', Proceedings of ICED'11, Vol. 2, pp. 344–355

[6] Rajarishi, S., Vei-Chung, L., Paredis, C.J.J. and Khosla, P. K. (2001) 'Modeling and simulation methods for design of engineering systems', Journal of Computing and Information Science in Engineering, Vol. 1 (1)

[7] Eisenbart, B., Dohr, F., Gericke, K., Vielhaber, M. and Blessing, L. (2013) 'Potentials for realising a consistent transition between function modelling with the IFM framework and early system simulation', Proceeding of the 19th international conference on engineering design – ICED'13

[8] Gausemeier, J., Frank, U., Donoth, J., and Kahl, S. (2009) 'Specification Technique for the Description of Self-optimizing Mechatronic Systems', Research in Engineering Design, Vol. 20 (4), pp. 201-223.

[9] Stark, R., Beier, G., Wöhler, T., and Figge, A. (2010) 'Cross-Domain Dependency Modelling – How to achieve consistent System Models with Tool Support', Proceedings of the 7th European Systems Engineering Conference, Stockholm.

[10] Eisenbart, B. (2014) 'Supporting Interdisciplinary System Development Through Integrated Function Modelling', Dissertation, Engineering Design and Methodology Group, University of Luxembourg, 2014

[11] Eisenbart, B., Qureshi, A., Gericke, K., Blessing, L.T.M. (2013) 'Integrating Different Functional Modelling Perspectives', in Global Product Development, ICoRD'13, Chakrabarti, A. and Prakash, R. (Eds.), Springer, ISBN 978-8132210498, pp. 85-97

[12] Eisenbart, B., Gericke, K., Blessing, L.T.M. (2014) 'Application of the IFM Framework for Modelling and Analysing System Functionality', Proceedings of the 13th International Design Conference – DESIGN, accepted for publication

[13] http://www.scenarioplus.org.uk, last visit: 2014-05-07

[14] Verein Deutscher Ingenieure, VDI guideline 3633 – Simulation of Systems in Materials Handling, Logistics and Production', Beuth Verlag, Duesseldorf

[15] Pahl G, Beitz W, Feldhusen J and Grote K H (2007) "Engineering Design – A Systematic Approach", Springer, Berlin, Heidelberg, New York, London.

[16] Weber C, Werner H, Deubel T. (2003) 'A different view on Product Data Management/Product Life-Cycle Management and its future potentials', J Eng Design 2003; 14:4 447-464